|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_04 | 학번 : | 20147351 | 이름 : | 정지현 |

1. **실행결과 확인**
2. 정수형 배열의 동적 할당 및 반환

|  |
| --- |
| #include <iostream>  using namespace std;  int main(){  cout<<”입력할 정수의 개수는?”;  int n;  cin>>n;  if(n<=0) return 0;  int \*p=new int[n]; //n개의 정수 배열 동적 할당  if(!p){  cout<<”메모리 할당 불가”’  return 0;  }  for(int i=0; i<n;i++){  cout<<i+1<<” 번째 정수 : “;  cin>>p[i];  }  int sum=0;  for(int i=0; i<n;i++)  sum += p[i];  cout<<”평균 : << sum/n<<endl;  delete[] p; //배열 메모리 반환  } |
| **[실행결과]** |

1. 프로그램 과제 1 : 교재 211 페이지 4번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include<cstring>  using namespace std;  class Sample {  int \*p; // 포인터 객체 p  int size;  public:  Sample(int n) {//생성자  size = n;  p = new int[n]; // n개 정수 배열의 동적 생성  }  void read();//동적 할당받은 정수 배열 p에 사용자로부터 정수를 입력 받음.  void write();//정수 배열을 화면에 출력  int big();//정수 배열에서 가장 큰 수 리텀  ~Sample() { delete[] p; }//소멸자  };  void Sample::read() {  for (int i = 0; i < size; i++) {  cin >> p[i];  }  }  void Sample::write() {  for (int i = 0; i < size; i++) {  cout << p[i] << ' ';  cout << endl;  }  }  int Sample::big() {  int big = p[0];  for (int i = 1; i < size; i++)  if (p[i] > big)  big = p[i];  return big;  }  int main() {  Sample s(10);  s.read();  s.write();  cout << " 가장 큰 수는 " << s.big() << endl;  char ch;  ch = getchar();  ch = getchar();  } |
| **[실행 결과]** |

1. 객체 배열의 동적 생성 및 반환

|  |
| --- |
| #include <iostream>  using namespace std;  class Circle {  int radius;  public:  Circle();  Circle(int r);  ~Circle();  void setRadius(int r) { radius = r; }  double getArea();  };  double Circle::getArea() {  return 3.14\*radius\*radius;  }  Circle::Circle() {  radius=1;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::Circle(int r) {  radius=r;  cout<<”생성자 실행 radius = “<<radius<<endl;  }  Circle::~Circle() {  radius=1;  cout<<”소멸자 실행 radius = “<<radius<<두이;  }  int main() {  Circle \*circleArray = new Circle[3]; // Circle 객체 배열 동적 생성, default 생성자 호출  circleArray[0].setRadius(10); circleArray[1].setRadius(20);  circleArray[2].setRadius(30);  for(int i=0; i<3; i++) // 배열의 각 원소 객체의 멤버 접근  cout << "Circle " << i << "의 면적은 " << circleArray[i].getArea() << endl;  Circle \*p= circleArray; // 포인터 p에 배열의 주소값 설정  for(int i=0; i<3; i++) { // 객체 포인터로 배열 접근  cout << "Circle " << i << "의 면적은 " << p->getArea() << endl;  p++; }  delete[] circleArray; //객체 배열 반환  } |
| **[실행결과]** |

1. 프로그램 과제 2 : 교재 213 페이지 8번 – 위의 예제로 제시된Circle 클래스를 사용할 것

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <cstring>  using namespace std;  class Circle {  int radius;  public:  void setRadius(int r) {  radius = r;  }  double getArea() {  return 3.14 \* radius\*radius;  }  };  int main() {  int n, m;  cout << "원의 개수 >> ";  cin >> n;  Circle \*p = new Circle[n];  for (int i = 0; i < n; i++) {  cout << "원" << i + 1 << "의 반지름 >> ";  cin >> m;  p[i].setRadius(m);  }  int count = 0;  for (int j = 0; j < n; j++) {  if (p[j].getArea() > 100) {  count++;  }  }  cout << "면적이 100보다 큰 원은" << count <<  "개 입니다." << endl;  delete[] p;  char ch;    ch = getchar();  ch = getchar();  } |
| **[실행 결과]** |